Rainfall Prediction in Tengger, Indonesia Using Hybrid Tsukamoto FIS and Genetic Algorithm Method

Ida Wahyuni* **& Wayan Firdaus Mahmudy**

Faculty of Computer Science, Brawijaya University,
8 Veteran Road, Malang, 65145, Indonesia
*E-mail: ida.wahyuni8@gmail.com

Abstract. Countries with a tropical climate, such as Indonesia, are highly dependent on rainfall prediction for many sectors, such as agriculture, aviation, and shipping. Rainfall has now become increasingly unpredictable due to climate change and this phenomenon also affects Indonesia. Therefore, a robust approach is required for more accurate rainfall prediction. The Tsukamoto Fuzzy Inference System (FIS) is one of the algorithms that can be used for prediction problems, but if its membership functions are not specified properly, the prediction error is still high. To improve the results, the boundaries of the membership functions can be adjusted automatically by using a genetic algorithm. The proposed genetic algorithm employs two selection processes. The first one uses the Roulette wheel method to select parents, while the second one uses the elitism method to select chromosomes for the next generation. Based on this approach, a rainfall prediction experiment was conducted for Tengger, Indonesia using historical rainfall data for ten-year periods. The proposed method generated root mean square errors (RMSE) of 6.78 and 6.63 for the areas of Tosari and Tutur respectively. These results are better compared with the results using Tsukamoto FIS and the Generalized Space Time Autoregressive (GSTAR) model from previous studies.

Keywords: *genetic algorithm; hybrid; prediction; rainfall; roulette wheel; Tsukamoto FIS.*

1 Introduction

Indonesia is a tropical country that has two seasons: dry and rainy [1]. Countries that have tropical climates depend on rainfall prediction, especially during the rainy season. This is very important given that there are many sectors that need reliable rainfall prediction data, including agriculture and transportation (notably air transport and sea transport) [2]. However, with climate change rainfall has become more difficult to predict. This also occurs in Indonesia, where it has become difficult to predict rainfall with the usual methods of weather forecasting. The challenge now is to build a system for rainfall prediction with as few parameters as possible using a simple mathematical model that can give reliable prediction results and few errors [3].

Received July 22nd, 2016, 1st Revision December 14th, 2016, 2nd Revision February 9th, 2017, Accepted for publication February 20th, 2017.
 ISSN: 2337-5787, DOI: 10.5614/itbj.ict.res.appl.2017.11.1.3

Rainfall prediction methods that exist today require many parameters, while the most influential parameters, especially for countries with a tropical climate, are atmospheric phenomena such as El Niño-Southern Oscillation (ENSO) and tropical cyclones [1]. As mentioned before, the climate change that has occurred in recent years has led to the phenomenon of atmospheric change, thus causing uncertainty in rainfall patterns.

The problem that must be resolved is how to create a rainfall prediction method with as few parameters as possible, which can give reliable prediction results and few errors. Rainfall prediction using one data type has been undertaken by Iriany, *et al.* [4]. The data used for the prediction of rainfall were rainfall data from the last ten years for Tengger. Application of the Generalized Space Time Autoregressive-Seemingly Unrelated Regression (GSTAR-SUR) method resulted in RMSE for the Tutur area of 10.89. Research using similar data was conducted by Wahyuni, Mahmudy, and Iriany [5] for the same location but with a different method. The algorithm was used Tsukamoto Fuzzy Inference System (FIS) resulting in an RMSE for the Tutur area of 8.64. Based on the results of this research, Tsukamoto FIS was found to be the better prediction algorithm. In the study conducted by Wahyuni, Mahmudy, and Iriany [5], the limits of the fuzzy membership functions were still determined manually so that the obtained results did not improve, despite showing a better RMSE than previous studies. The fuzzy membership functions from this research should be optimized to obtain better prediction results. Optimization can be done using meta-heuristic algorithms that have been proven successful when applied to complex problems [6].

A meta-heuristic algorithm often used for optimization problems is the genetic algorithm (GA). Genetic algorithms have been used to optimize the boundaries of fuzzy membership functions by several researchers. Zhang, *et al.* [7] optimized the boundaries of the fuzzy membership functions with a real coded genetic algorithm (RCGA). Cazarez-Castro, *et al.* [8] optimized the boundaries of the fuzzy membership functions with a genetic algorithm. Genetic algorithms work by modeling the constraints on a membership function of fuzzy input parameters into chromosomes. These chromosomes are put through the process of reproduction and selection to select the chromosomes with the most optimal fitness value. The chromosomes with the most optimal fitness value will then be used as limitations for the fuzzy membership function. The most optimal fitness value is determined by the counting process of Tsukamoto FIS applied in the stage of fitness calculation.

A combination of the Fuzzy Optimal Model (FOM) and a GA was used by Cheng and Chau [9] to solve the multi-objective rainfall-runoff model calibration. This method was able to solve the multiple-objective run-off routing parameter calibration problem. The model used in the research of Cheng and Chau showed that the hybrid FOM and GA was capable of further exploring the important characteristics of floods robustly and efficiently.

Thus, in this paper a hybrid method using Tsukamoto FIS and GA for rainfall prediction is proposed. The data used in the experiment for the prediction of rainfall are the rainfall data from Tengger area over 2005 to 2014 in *dasarian* (data covering ten days). The parameters that were used as input parameters in Tsukamoto FIS were: rainfall data from 10 days prior (Zt-1), rainfall data from 20 days prior (Zt-2), rainfall data from 170 days prior (Zt-17), and rainfall data from 340 days prior (Zt-34). The rainfall data and input parameters used were similar to those used in the previous study conducted by Wahyuni, Mahmudy, and Iriany [5]. This was done to see whether or not the results of rainfall prediction with membership function optimization using a genetic algorithm produced better predictions.

2 Tsukamoto Fuzzy Inference System

The Tsukamoto FIS method is a computational framework that is based on fuzzy reasoning, fuzzy rules in the form of if-then rules, and fuzzy set theory [10]. The general method of Tsukamoto FIS comprises several processes, i.e. fuzzy set establishment, fuzzy inference rule application, and defuzzification [11]. The main difference between Tsukamoto FIS and typical fuzzy logic is in the defuzzification process, where the process is done to calculate the value of the outputs in the form of crisp values (z). Defuzzification is calculated with the center average defuzzifier formula shown in Eq. (1) [5].

$$Z = \frac{\sum(\alpha\_p_i * Z_i)}{\sum \alpha\_p_i} \tag{1}$$

Description:

Z : centralized average

α_p : alpha predicate value (minimum value of the degree of membership)

Zi : crisp values obtained from the results of inference

i : number of fuzzy rules

3 Hybrid Tsukamoto FIS and Genetic Algorithm

The Tsukamoto fuzzy inference system requires appropriate boundaries of its membership functions to obtain accurate prediction results. The boundaries can be determined automatically using a genetic algorithm. Genetic algorithms have several stages of problem solving, i.e. representation of chromosomes, initialization of the population, calculation of the value of fitness, reproduction (including crossover and mutation), and selection.

3.1 Chromosome Representation

A chromosome is a representation of a solution that is encoded in genes [12]. In this study, the chromosomes are formed by a real coded genetic algorithm (RCGA). Real coded genetic algorithms (RCGA) use an array containing real numbers as a representation of chromosomes [13,14] to limit membership functions. In one chromosome there are eleven genes. Every existing gene on one chromosome represents the boundaries of the membership functions for each input criterion to model the Tsukamoto fuzzy inference system. The first and second genes are used to represent the boundaries of the membership function for the first criteria. The third and fourth genes are used to represent the boundaries of the membership function for the second criteria. The fifth and sixth genes are used to represent the boundaries of the membership function for the third criteria. The seventh and eighth genes are used to represent the boundaries of the membership function for the fourth criteria. The eighth up to the eleventh genes are used to represent the boundaries of the membership function for the result criteria. The representation of the chromosomes is shown in Figure 1.

C1	a1	a2	a3	a4	a5	a6	a7	a8	a9	a10	a11
	Zt-1		Zt-2		Zt-17		Zt-38			Zt	

Figure 1 Illustration of chromosome representation.

where:

Ci	: chromosomes to *i*
a1, a2	: segment boundaries of membership functions for Zt-1
a3, a4	: segment boundaries of membership functions for Zt-2
a5, a6	: segment boundaries of membership functions for Zt-17
a7, a8	: segment boundaries of membership functions for Zt-34
a9, a10, a11	: segment boundaries of membership functions for Zt

An example of the chromosome representation using real code is shown in Figure 2. After the chromosomes are represented randomly from the range [0, 40], the values of the genes on the chromosomes are sorted in ascending order. The gene values are sorted according to each segment, consisting of two genes. The results of chromosome sorting are shown in Figure 3.

C1	35.1	19.5	1.0	6.0	36.1	26.2	17.4	14.3	21.31	7.422	14.3

Figure 2 Example of chromosome representation.

C1	19.5	35.1	1.0	6.0	26.2	36.1	14.3	17.4	7.422	14.3	21.31

Figure 3 Example of chromosome representation after ordering.

3.2 Initialization of Population

Initialization of the population is the stage that is performed first during the process of running the genetic algorithm. The total population of the genetic algorithm is often called the *popSize* variable. For example, the population will be shaped by three chromosomes or can be the size of a population of 3 from C1, ... Cn, where n = 3. The initialization of the population is shown in Table 1.

Table 1 Example of chromosome initialization.

C	G1	G2	G3	G4	G5	G6	G7	G8	G9	G10	G11
C1	19.5	35.1	1.0	6.0	26.2	36.1	14.3	17.4	7.42	14.3	21.31
C2	18.2	30.2	2.2	8.2	16.5	28.1	2.4	17.6	8.76	23.13	35.49
C3	8.2	24.9	13.1	26.7	10.0	14.1	3.2	29.4	4.87	16.0	19.25

3.3 Calculation Fitness Function

The fitness function is used to measure how good an obtained solution is. The process of calculating the value of fitness in this case was done by using the Tsukamoto fuzzy inference system. By using the Tsukamoto fuzzy inference system, the chromosome that produces the smallest error is determined. The best chromosome is the one that has the smallest error value. The goal of this research was to find the smallest prediction error for the calculation of the fitness (f) value expressed in Eq. (2). To determine the error, the RMSE formula expressed in Eq. (3) was used. In the RMSE calculation process, the error value between the actual rainfall data and the predicted results from the rainfall data must be known. The error formula is expressed in Eq. (4).

$$f = \frac{1}{e} \tag{2}$$

where

f : fitness
e : RMSE error

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - y_i')^2} \tag{3}$$

where

RMSE : root mean square error
yi' : predicted data
yi : actual data

n : number of data

$$Error = y - y' \quad (4)$$

where:

y : actual data
y' : predicted data

The process of calculating the Tsukamoto fuzzy inference system was described in Wahyuni, Mahmudy, and Iriany [5]. The fitness calculation on chromosome 1 is shown in Table 2.

Table 2 Illustration of fitness calculate.

C	Value of Gene							Area	RMSE	Sum of RMSE	(*f*)
C1	19.5	35.1	1	…	7.42	14.3	21.3	Puspo	7.751	31.112	0.032
								Sumber	8.071		
								Tosari	7.606		
								Tutur	7.684		
C2	18.2	30.2	2.2	…	8.76	23.1	35.4	Puspo	10.57	42.336	0.024
								Sumber	10.87		
								Tosari	10.42		
								Tutur	10.47		
C3	8.2	24.9	13.1	…	4.87	16	19.2	Puspo	7.361	28.657	0.035
								Sumber	7.453		
								Tosari	7.033		
								Tutur	6.81		

3.4 Selection

The most important process in the genetic algorithm is the process of reproduction, and thus a selection process is needed to choose the parents involved in the process [15]. There are many selection methods, including random selection, rank selection, tournament selection, stochastic universal sampling, Boltzmann selection, and Roulette wheel selection [16]. The method of selection that was used in this research was Roulette wheel, because this method can choose parents that qualify for the reproductive process by calculating the probability value of each parent [17]. The initial step in Roulette wheel selection is calculating the probability and the cumulative value of each chromosome. Then, a random value is selected between [0, 1] to determine the chromosome that will be the parent, with the provisions of $C[n-1] < R < C[n]$. Calculation of the probability value of each chromosome is shown in Table 3 and the Roulette wheel selection results are shown in Table 4.

Table 3 Probability and cumulative calculation results for each individual.

C	Fitness (*f*)	Probability	Cumulative
C1	0.032	0.351648352	0.351648352
C2	0.024	0.263736264	0.615384615
C3	0.035	0.384615385	1

Table 4 Result of Roulette Wheel selection.

Random	Chromosome
0.07720	C1
0.25347	C2

3.5 Crossover

The process of crossover is one of the reproduction processes in the genetic algorithm. Reproduction in genetic algorithms consists of the crossover and mutation processes. The parents used in the crossover process are two chromosomes selected by the Roulette wheel selection in the previous stage.

In this study, the method of crossover used was extended intermediate crossover [13]. This process produces crossover offspring, where the number of offspring is derived from the formula offspring = *cr* × *popSize*. The variable *cr*, or crossover rate, is one of the parameters in a genetic algorithm [18]. The value of the crossover rate is previously determined within the value range [0, 1]. Suppose, the *cr* to be used is 0.5, then the number of generated offspring is, for example: offspring = *cr* × *PopSize* = 0.5 × 3 = 2. The method of extended intermediate crossover begins by selecting the parents P1 = (p11, … pn1) and P2 = (p12, … pn2). Offspring O = (O1, … O2) are produced by the formula in Eq. (5). The step-by-step process of extended intermediate crossover is shown in Figure 4.

$$O_i = p_i{}^2 + \alpha_1(p_i{}^2 - p_i{}^1) \quad (5)$$

where:

O : offspring
p : parent
α : random value range [-0.25, 1.25]

C1	19.5	35.1	1.0	6.0	26.2	36.1	14.3	17.4	7.422	14.3	21.31
C2	18.2	30.2	2.2	8.2	16.5	28.1	2.4	17.6	8.758	23.13	35.49
α	0.59	0.489	0.632	0.814	0.57	0.473	0.251	-0.19	0.912	0.283	-0.2
O1	18.73	32.7	1.759	7.79	20.67	32.31	11.31	17.36	8.641	16.8	18.48

Figure 4 Example of extended process intermediate crossover.

3.6 Mutation

The second stage of the reproduction process is mutation. In this study simple random mutation was employed. The simple random mutation process begins by determining offspring O = (O1, … O2) by Eq. (6) from parent P = (p1, … pn) [1]. An example of a simple random mutation process is shown in Figure 5. The variable *mr*, or the mutation rate, had a value of 0.1, which was predetermined from the value range [0, 1]. The process of mutation will produce offspring with offspring formula = *mr* x *popSize* = 0.1 x 3 = 0.3 = 1 offspring.

$$O_i = p_i(1 + \alpha) \tag{6}$$

where:

O : offspring
p : parent
α : random value from the interval [-0.1, 0.1]

C1	19.5	35.1	1.0	6.0	26.2	36.1	14.3	17.4	7.4	14.3	21.3
A	-0.03	0.02	0.03	0.04	0.05	0.04	-0.10	0.01	0.04	0.09	0.10
O2	18.9	35.9	1.0	6.2	27.6	37.4	12.9	17.5	7.7	15.6	23.4

Figure 5 Examples of simple random mutation.

3.7 Evaluation

After the reproduction process is completed, the reproduction fitness values of parent and offspring are calculated according to Eq. (2). Chromosomes with a higher fitness value have a higher probability to be passed onto the next generations. The results of the fitness calculation after the reproduction process are shown in Table 5.

Table 5 Calculation of fitness results after reproduction process.

Parent	Fitness (*f*)
C1	0.032
C2	0.024
C3	0.035
O1	0.030
O2	0.031

3.8 Elitism

After the reproductive process and evaluation are complete, parent and offspring reproductions undergo a second process of selection. The second selection process is used to choose the chromosomes that will pass to the next generation, replacing the chromosomes in the old population [16]. The method of selection that is used in the proposed method is elitism, which chooses the chromosomes to pass to the next generation by sorting the fitness value ranging from the largest to the smallest. As defined, the total number of chromosomes that are selected for elitism selection is equal to the population size. An example of the chromosome with fitness value and elitism selection results is shown in Table 6.

Table 6 Result of Elitism selection.

Chromosome	Fitness
C3	0.035
C1	0.032
O2	0.031

3.9 Stopping Condition

Genetic algorithms do not indicate when the iteration process has found the best solution. However, a variety of methods can be used to determine at which state the iterations of the genetic algorithm should stop [19]. There are two types of conditions used for the stopped state, assuming that the population has produced an optimal solution [20]. The first is to take a regeneration limit value, for example 100 generations. The second is to calculate the member replacement failure of the population occurring in the sequence; it is assumed that there are 10 chromosome replacement failures in one generation cycle.

4 Experiment and Result

An experiment was conducted using four locations: Puspo, Sumber, Tosari, and Tutur. These four locations are located in the area of Tengger, East Java. The data used for the prediction were rainfall data from previous years, from 2005 to 2013. 326 sets of rainfall data were used as the reference for input. Each record represented rainfall for ten days, so that the predicted outcomes would generate rainfall data for the next ten days. Four pieces of data were used as criteria for input, i.e. Zt-1 or rainfall from 10 days prior, Zt-2 or rainfall from 20 days prior, Zt-17 or rainfall from 170 days prior, and Zt-34 or rainfall from 340 days prior.

In the testing process, the data were tested on the genetic algorithm parameters used to optimize the boundaries of the membership functions. The testing included testing of the population size, the size of the generation, and the combination of *cr* and *mr*.

4.1 Testing Population Size

The process of testing the population size was done five times; the size of the population was 50 to 250 with increments of 50. The values of *cr* and *mr* used at the start were 0.9 and 0.1, while the number of first generations used was

100. The resulting fitness values from five tests were taken for calculation of the average. From the average, the most optimal population size was known. The test results on the population size are shown in Table 7. The chart of fitness value changes according to population size can be seen in Figure 6.

Table 7 Result of testing population size.

Population Size	Fitness Value					Fitness Average
	Number of Testing					
	1	2	3	4	5	
50	0.030	0.026	0.034	0.034	0.027	0.030
100	0.034	0.032	0.031	0.032	0.034	0.032
150	0.032	0.033	0.033	0.032	0.034	0.033
200	0.032	0.033	0.035	0.034	0.032	0.033
250	0.034	0.033	0.032	0.033	0.034	0.033

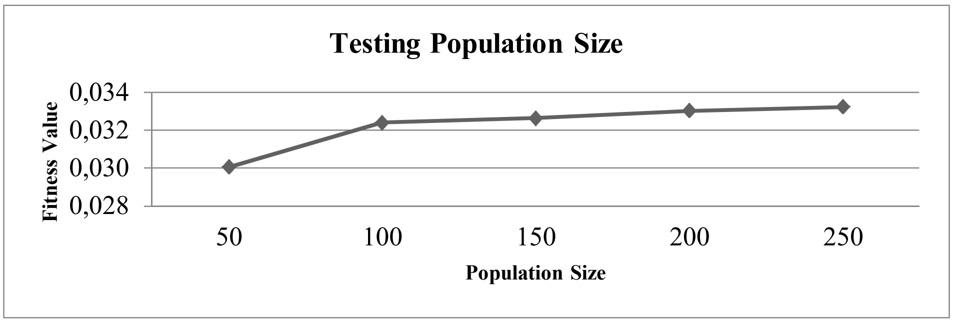

Figure 6 Chart of testing population size.

The results of the testing population size in Figure 6 indicate that on average the best fitness was obtained with a population size of 200. For this population size, the highest fitness value obtained was 0.035 and there was no significant increase in this value for larger population sizes. In a previous study it has also been observed that the fitness value increases in proportion to the size of the population, but after a certain point of the rise in population size, the increase in fitness is not significant [14].

4.2 Testing Generation Size

The testing process for generation size was done five times; the size of the population was 100 to 500 with increments of 100. The values of *cr* and *mr* used at the start were 0.9 and 0.1, while the population size used was 200, which yielded the best fitness according to the test results for population size. The average fitness value resulting from the test was counted five times. The average result allowed the most optimal generation size to be known. The test results for generation size are shown in Table 8. The chart of fitness value changes according to generation size can be seen in Figure 7.

Table 8 Testing results for generation size testing.

Generation Size	Fitness Value					Fitness Average
	Number of Testing					
	1	2	3	4	5	
100	0.032	0.033	0.035	0.034	0.032	0.033
200	0.033	0.032	0.032	0.032	0.034	0.033
300	0.033	0.035	0.035	0.033	0.032	0.033
400	0.032	0.032	0.033	0.031	0.035	0.032
500	0.035	0.033	0.032	0.033	0.032	0.033

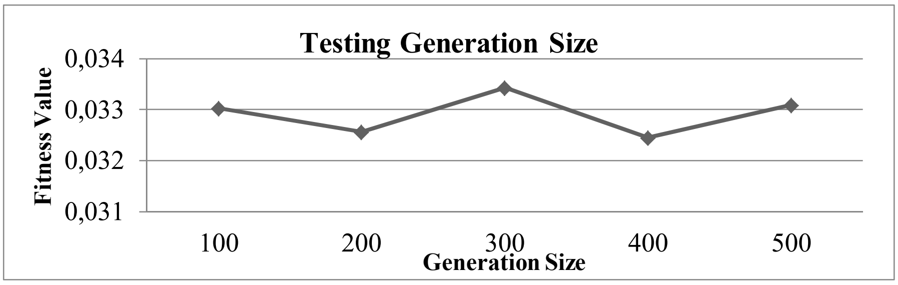

Figure 7 Chart of generation size testing.

The results of generation size testing in Figure 7 indicate that the best fitness average was obtained at a generation size of 300. The obtained fitness value for this generation size was 0.035 and there was no significant increase in this value for larger generation sizes. In a previous research it has also been observed that higher obtained fitness values are not too significant for generation sizes greater than 100 [20].

4.3 Testing *cr* and *mr* Value

The testing process of *cr* and *mr* combinations was based on three theories, namely the theories of Grefenstette [21], De Jong, and Theory of Fitness Quality Monitoring, to determine the *cr* value. The best population size and generation size were used in testing *cr* and *mr*. Tests were performed five times and the values of fitness resulting from the five tests were used for calculation of the average. From the average yield, the most optimal sizes of *cr* and *mr* were known. The test results for population size are shown in Table 9. The chart of changes of the *cr* and *mr* values in accordance with the fitness value can be seen in Figure 8.

Table 9 Testing results of *cr* and *mr* combinations.

Value of cr	Fitness Value					Fitness Average
	Number of Testing					
	1	2	3	4	5	
0.45	0.031	0.030	0.032	0.028	0.036	0.031
0.6	0.034	0.035	0.036	0.032	0.028	0.033
0.95	0.033	0.036	0.036	0.032	0.033	0.034

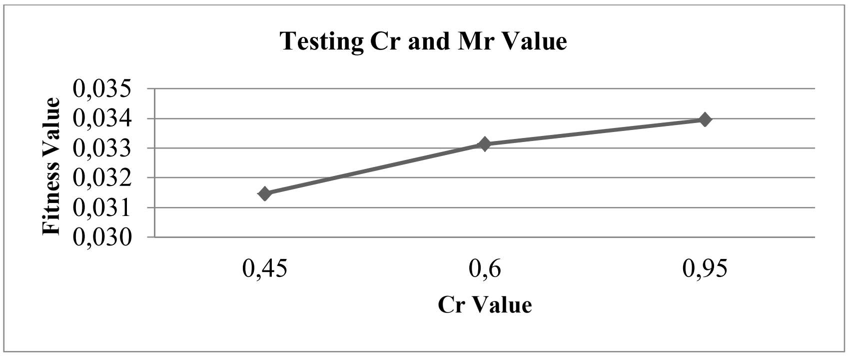

Figure 8 Chart of testing *cr* and *mr* values.

Based on the test sizes of *cr* and *mr,* it was found that on average the best fitness was found for the combination of *cr* at 0.95 and *mr* at 0.05. Combining *cr* and *mr*, the highest obtained fitness value was 0.0356. This fitness value was compared with the best fitness values obtained from other combinations. The values for the smaller *cr* fitness results were not optimal because the genetic algorithm using random search would not be able to explore the search for solutions effectively.

4.4 Boundaries of Membership Function Optimization Results

Based on the genetic algorithm parameter testing processes that were done, it was found that the optimal parameter values were a population size of 100, a generation size of 300, and a combination of *cr* at 0.95 and *mr* at 0.05. The solutions resulting from the calculation of the genetic algorithm for which the chromosomes had the best fitness value were used as constraints on the Tsukamoto FIS membership function. The optimized boundaries of the membership functions are shown in Figures 9 to 13.

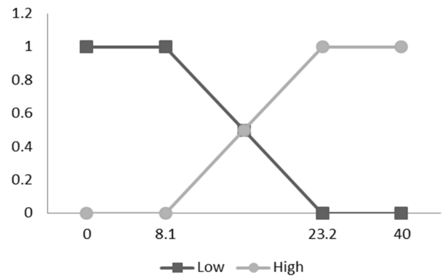

Figure 9 Boundaries of membership functions for Zt-1.

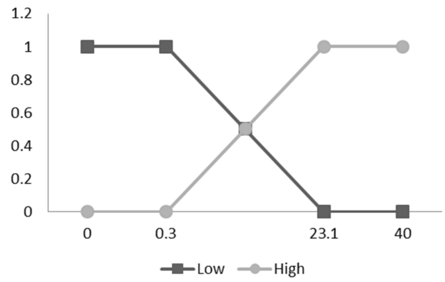

Figure 10 Boundaries of membership functions for Zt-2.

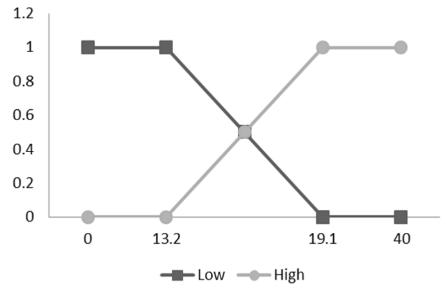

Figure 11 Boundaries of membership functions for Zt-17.

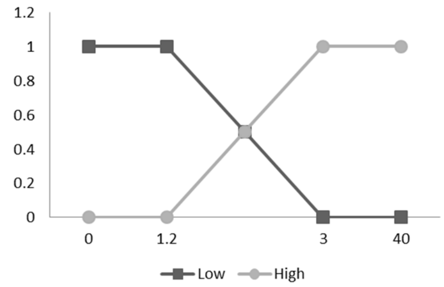

Figure 12 Boundaries of membership functions for Zt-34.

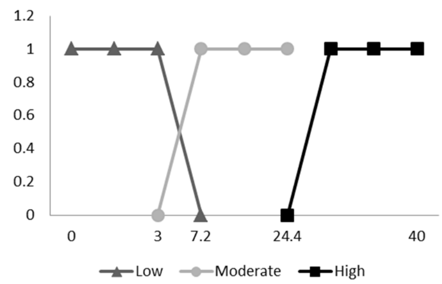

Figure 13 Boundaries of membership functions for Zt.

4.5 Calculation of Accuracy of Prediction Results

The results of rainfall prediction using the proposed hybrid Tsukamoto FIS and GA method are shown in Table 10. The tables also show the error count results using Eq. (4). The error value produced was used to calculate the value of RMSE using Eq. (3).

Table 10 Results of rainfall prediction using Hybrid Tsukamoto FIS and Genetic Algorithm method in Tutur area.

Rainfall Prediction Result in Tutur Area			
Number	**Actual rainfall data (y)**	**Prediction result (y')**	**Error**
1	12.600	8.170662471	4.429337529
2	10.900	7.377007007	3.522992993
3	9.364	9.775355248	-0.411718884
4	14.500	8.698977019	5.801022981
5	12.000	5.928947368	6.071052632
6	5.875	10.40865203	-4.533652033
7	16.200	4.217447368	11.98255263
8	5.700	4.068421053	1.631578947
9	6.909	4.510526316	2.398564593
10	6.100	6.901928643	-0.801928643
…	…	…	…
318	3.091	5.014519852	-1.92361076
319	2.400	9.573894785	-7.17389478
320	14.000	11.65408864	2.345911364
321	20.500	8.698977019	11.80102298
322	12.000	11.6769736	0.323026402
323	20.600	9.864538392	10.73546161
324	14.727	10.00798978	4.719282946
325	15.100	5.691131579	9.408868421
326	5.200	9.935013683	-4.73501368

4.6 Comparison of Prediction Results

The results of the rainfall prediction using the hybrid Tsukamoto FIS and GA method were compared with the prediction results from previous studies conducted by Wahyuni, Mahmudy, and Iriany [2] and Iriany, *et al.* [3]. The standard of comparison used was the predicted RMSE value for each location. The comparison is shown in Table 11. From Table 11 it can be seen that the result of rainfall prediction using the hybrid method produced lower predicted RMSE values for the four areas of Puspo, Sumber, Tosari, and Tutur when compared with the results predicted by the Tsukamoto FIS algorithm alone. The hybrid method also generated a smaller RMSE at Tosari and Tutur area when compared with the prediction results using the GSTAR-SUR method. The RMSE difference in the Sumber area using the GSTAR-SUR method was only better by 0.4 than the RMSE using the hybrid method. However, the predictions showed that the hybrid method generally produced better predictions indicated by the smaller RMSE values.

Table 11 Comparison of prediction results with Tsukamoto FIS and GSTAR methods.

Number	Location	RMSE Hybrid Fuzzy GA	RMSE Fuzzy	RMSE GSTAR
1	Puspo	7.30	8.95	4.9
2	Sumber	7.09	9.64	6.69
3	Tosari	6.78	8.81	7.92
4	Tutur	6.63	8.64	10.89

5 Conclusions

Based on the results of this study, the proposed hybrid Tsukamoto FIS and GA method produced lower RMSE values when compared with the rainfall predictions using the Tsukamoto FIS and GSTAR-SUR methods. Only the RMSE value at Puspo was higher compared to that produced by the GSTAR-SUR method. This indicates that optimizing the boundaries on the Tsukamoto FIS membership functions effectively improves prediction results. This model will be further developed for research in other locations in Java. Additional optimization needs to be done on the Tsukamoto FIS modeling in order to produce lower RMSEs. In addition to restriction of the membership functions, the constraints on Tsukamoto FIS could also be optimized [22]. Optimizing the fuzzy rules can be done by several methods: Formal Concept Analysis Method [22], Multi-Objective Genetic Algorithms [23], or Fuzzy Boolean Nets [24], or Artificial Neural Network (ANN) [25].

Acknowledgements

This research was supported by the local Meteorological and Geophysics Agency Karangploso, East Java, Indonesia and Faculty of Computer Science, Brawijaya University.

References

[1] Annas, S., Kanai, T. & Koyama, S., *Assessing Daily Tropical Rainfall Variations using a Neuro-Fuzzy Classification Model*, Ecological Informatics, **2**(2), pp. 159-166, 2007.

[2] Kashid, S.S. & Maity, R., *Prediction of Monthly Rainfall on Homogeneous Monsoon Regions of India based on Large Scale Circulation Patterns using Genetic Programming*, J. Hydrol., **454-455**, pp. 26-41, 2012.

[3] Chang, F.J., Chiang, Y.M., Tsai, M.J., Shieh, M.C., Hsu, K.L. & Sorooshian, S., *Watershed Rainfall Forecasting using Neuro-fuzzy Networks with the Assimilation of Multi-sensor Information*, J. Hydrol., **508**, pp. 374-384, 2014.

[4] Iriany, A., Mahmudy, W.F., Sulistyono, A.D. & Nisak, S.C., *GSTAR-SUR Model for Rainfall Forecasting in Tengger Region, East Java*, 1st Int. Conf. Pure Appl. Res. Univ. Muhammadiyah Malang, 21-22 August, **1**, pp. 1-8, 2015.

[5] Wahyuni, I., Mahmudy, W.F. & Iriany, A., *Rainfall Prediction in Tengger Region-Indonesia Using Tsukamoto Fuzzy Inference System*, 1st Int. Conf. Inf. Technol. Inf. Syst. Electr. Eng., **1**, pp. 1-11, 2016.

[6] Mahmudy, W.F., Marian, R.M. & Luong, L.H.S., *Hybrid Genetic Algorithms for Multi-Period Part Type Selection and Machine Loading Problems in Flexible Manufacturing System*, IEEE Int. Conf. Comput. Intell. Cybern. Yogyakarta, Indonesia. 3-4 December, **13**(12), pp. 126-130, 2013.

[7] Zhang, H., Wang, F. & Zhang, B.O., *Genetic Optimization of Fuzzy Membership Functions*, Proc. 2009 Int. Conf. Wavelet Anal. Pattern Recognition, Baoding, July, **9**(7), pp. 465-470, 2009.

[8] Cazarez-Castro, N.R., Aguilar, L.T. & Castillo, O., *Fuzzy Logic Control with Genetic Membership Function Parameters Optimization for the Output Regulation of a Servomechanism with Nonlinear Backlash*, Expert Syst. Appl., **37**(6), pp. 4368-4378, 2010.

[9] Cheng, C.T., Ou, C.P. & Chau, K.W., *Combining a Fuzzy Optimal Model with a Genetic Algorithm to Solve Multi-objective Rainfall-runoff Model Calibration*, J. Hydrol., **268**(1-4), pp. 72-86, 2002.

[10] Sasmito, G.W. & Somantri, O., *Tsukamoto Method in Decision Support System for Realization of Credit on Cooperative*, 4th ICIBA 2015, Int. Conf. Inf. Technol. Eng. Appl. Palembang, **4**, pp. 39-44, 2015.

[11] Sari, N.R. & Mahmudy, W.F., *Fuzzy Inference Tsukamoto System for Determining Eligibility for Prospective Employees*, Semin. Nas. Sist. Inf. Indones. 2-4 Nop. 2015, No. 2002, pp. 2-4, 2015. (Text in Indonesian)

[12] Ding, Y. & Fu, X., *Kernel-based Fuzzy C-means Clustering Algorithm based on Genetic Algorithm*, Neurocomputing, **188**, pp. 233-238, 2015.

[13] Mahmudy, W., Marian, R. & Luong, L.H.S., *Real Coded Genetic Algorithms for Solving Flexible Job-shop Scheduling Problem – Part I: Modeling*, Adv. Mater. Res., **701**, pp. 359-363, 2013.

[14] Mahmudy, W., Marian, R. & Luong, L.H.S., *Real Coded Genetic Algorithms for Solving Flexible Job-Shop Scheduling Problem – Part II: Optimization*, Adv. Mater. Res., **701**, pp. 364-369, 2013.

[15] Murata, T. & Ishibuchi, H., *Adjusting Membership Functions of Fuzzy Classification Rules by Genetic Algorithms*, Proc. 1995 IEEE Int. Conf.

Fuzzy Syst. Int. Jt. Conf. Fourth IEEE Int. Conf. Fuzzy Syst. Second Int. Fuzzy Eng. Symp., **4**, pp. 1819-1824, 1995.

[16] Moradi, S.T. & Nikolaev, N.I., *Optimization of Cement Spacer Rheology Model Using Genetic Algorithm*, IJE Trans. A Basics, **29**(1), pp. 127-131, 2016.

[17] Mahmudy, W.F., *Optimization of Part Type Selection and Loading Problem with Alternative Production Plans in Flexible Manufacturing System using Hybrid Genetic Algorithms – Part 2 : Genetic Operators and Results*, 2013 5th Int. Conf. Knowl. Smart Technol. Optim., **5**, pp. 81-85, 2013.

[18] Herman, N.S., *Genetic Algorithms and Designing Membership Function in Fuzzy Logic Controllers*, World Congr. Nat. Biol. Inspired Comput. (NaBIC 2009), **9**, pp. 1753-1758, 2009.

[19] Jafarian, J., *An Experiment to Study Wandering Salesman Applicability on Solving the Travelling Salesman Problem based on Genetic Algorithm*, Int. Conf. Educ. Inf. Technol. (ICEIT 2010) An, **10**, pp. 1-7, 2010.

[20] Wahyuni, I., Mahmudy, W.F. & Iriany, A. *Rainfall Prediction using Hybrid Adaptive Neuro Fuzzy Inference System (ANFIS) and Genetic Algorithm*, J. Telecommun. Electron. Comput. Eng., **9**, 2017 (Article in Press).

[21] Grefenstette, J., *Optimization of Control Parameters for Genetic Algorithms*, IEEE Trans. Syst. Man. Cybern., **16**(1), pp. 122–128, 1986.

[22] Cintra, M.E., Camargo, H.A. & Monard, M.C., *Genetic Generation of Fuzzy Systems with Rule Extraction using Formal Concept Analysis*, Inf. Sci. (Ny)., **349-350**, pp. 199-215, 2016.

[23] Ishibuchi, H., Murata, T. & Gen, M., *Performance Evaluation of Fuzzy Rule-based Classification Systems Obtained by Multi-objective Genetic Algorithms*, Comput. Ind. Eng., **35**(3-4), pp. 575-578, 1998.

[24] Carvalho, J.P. & Tomé, J., *Qualitative Optimization of Fuzzy Causal Rule Bases using Fuzzy Boolean Nets*, Fuzzy Sets Syst., **158**(17), pp. 1931-1946, 2007.

[25] Janeela Theresa, M.M. & Joseph Raj, V., *Fuzzy Based Genetic Neural Networks for The Classification of Murder Cases Using Trapezoidal and Lagrange Interpolation Membership Functions*, Appl. Soft Comput. J., **13**(1), pp. 743-754, 2013.